Australian Government
Department of Defence
Science and Technology

Evaluation of Available Software for Reconstruction of a Structure from its Imagery

Leonid K Antanovskii

Weapons and Combat Systems Division
Defence Science and Technology Group

DST-Group–TR–3356

ABSTRACT

In this report the *Computer Vision System* toolbox of MATLAB® and the *Visual Structure from Motion* software are evaluated on three datasets of airborne imagery, provided by Defence Research and Development Canada, and on a dataset of synthetic imagery generated by the VIRSuite software developed in the Defence Science and Technology Group. The user interface for a developed code for structure reconstruction, based on the feature detection algorithms of the MATLAB toolbox, is described.

RELEASE LIMITATION

Approved for public release

Published by

Weapons and Combat Systems Division
Defence Science and Technology Group
PO Box 1500
Edinburgh, South Australia 5111, Australia

Telephone: *1300 333 362*
Facsimile: *(08) 7389 6567*

April, 2017
AR-016-831

APPROVED FOR PUBLIC RELEASE

Evaluation of Available Software for Reconstruction of a Structure from its Imagery

Executive Summary

The objectives of the "Three-Dimensional Target Reconstruction" International Collaboration Project (CP 7–25–13) between Australia, Canada and USA, are to extract three-dimensional information from a target via low to high cost vision sensors on moving ground or air platforms through algorithm development. Methods of analysis include comparison of three-dimensional reconstruction algorithms to 'truth' data gathered via Laser Detection and Ranging (LADAR) sensors including inertial sensors.

Existing techniques allow reliable reconstruction of smooth surfaces, such as terrains, but the reconstruction of three-dimensional buildings is very challenging with automated software. This project is aimed at investigating new techniques to improve robustness of the reconstruction process, and also looks at the issues of using LADAR data for direct reconstruction. LADAR data is often sparse, particularly at longer ranges, and the fusion of two-dimensional imagery can be beneficial. The expected defence outcomes from the project are:

- the development of robust techniques for mission planning when using imaging seekers
- collaboration on algorithmic techniques for automatic target reconstruction, detection and recognition
- improved simulation techniques and sharing of test data sets.

This report partially addresses the objectives of the collaboration project. The *Computer Vision System* toolbox of MATLAB® and the *Visual Structure from Motion* (VisualSFM) software are evaluated on three datasets of airborne imagery, provided by Defence Research and Development Canada, and on a dataset of synthetic imagery generated by the VIRSuite software developed in the Defence Science and Technology Group. The user interface for a developed code for structure reconstruction, based on the feature detection algorithms of the MATLAB toolbox, is described.

THIS PAGE IS INTENTIONALLY BLANK

Author

Leonid K Antanovskii
Weapons and Combat Systems Division

Born in Siberia, Leonid Antanovskii holds a Master of Science (with distinction) in Mechanics and Applied Mathematics from the Novosibirsk State University and a PhD in Mechanics of Fluid, Gas and Plasma from the Lavrentyev Institute of Hydrodynamics of the Russian Academy of Science. Since graduation he worked for the Lavrentyev Institute of Hydrodynamics (Russia), Microgravity Advanced Research & Support Center (Italy), the University of the West Indies (Trinidad & Tobago), and in private industry in the USA and Australia.

Leonid Antanovskii joined the Defence Science and Technology Group in February 2007 working in the area of weapon–target interaction.

THIS PAGE IS INTENTIONALLY BLANK

Contents

List of Figures

List of Tables

THIS PAGE IS INTENTIONALLY BLANK

1 Introduction

The objectives of the "Three-Dimensional Target Reconstruction" International Collaboration Project (CP 7–25–13) between Australia, Canada and USA, are to extract three-dimensional information from a target via low to high cost vision sensors on moving ground or air platforms through algorithm development. Methods of analysis include comparison of three-dimensional reconstruction algorithms to 'truth' data gathered via Laser Detection and Ranging (LADAR) sensors including inertial sensors.

Existing techniques allow reliable reconstruction of smooth surfaces, such as terrains, but the reconstruction of three-dimensional buildings is very challenging with automated software. This project is aimed at investigating new techniques to improve robustness of the reconstruction process, and also looks at the issues of using LADAR data for direct reconstruction. LADAR data is often sparse, particularly at longer ranges, and the fusion of two-dimensional imagery can be beneficial. The expected defence outcomes from the project are:

- the development of robust techniques for mission planning when using imaging seekers
- collaboration on algorithmic techniques for automatic target reconstruction, detection and recognition
- improved simulation techniques and sharing of test data sets.

A series of reports have been published [Antanovskii 2014, Antanovskii 2016*b*, Antanovskii 2016*a*, Antanovskii 2017*b*, Antanovskii 2017*a*] which partially address the objectives of the collaboration project. In particular, a prototype MATLAB code for 3D target reconstruction is developed, which implements the basic 3D reconstruction algorithms described in [Hartley & Zisserman 2003]. In this follow-up report the *Computer Vision System* toolbox of MATLAB® and the *Visual Structure from Motion* (VisualSFM) software are evaluated on three datasets of airborne imagery, provided by Defence Research and Development Canada (DRDC), and on a dataset of synthetic imagery generated by the VIRSuite software developed in the Defence Science and Technology Group [Swierkowski et al. 2014]. The user interface for a developed code for structure reconstruction, based on the feature detection algorithms of the MATLAB toolbox, is described.

2 Overview of feature detection algorithms

Comprehensive background material and basic numerical methods for structure reconstruction are provided in [Hartley & Zisserman 2003]. It is worthwhile emphasizing that this monograph does not address the feature detection technology, which constitutes a broad subject of research in its own, and in most situations assumes that all world points of interest are visible in all views. In real-world scenarios this is not the case due to point occlusion.

Feature detection is a low-level image processing operation based on examining every pixel of an image and its immediate neighbourhood to associate feature descriptors to points of interest, called *keypoints*; the extracted descriptors are then used to match the keypoints in any two images. Several feature detection algorithms are publicly available, namely:

- *Combined Corner and Edge Detector* algorithm [Harris & Stephens 1988]

- *Minimum Eigenvalue* algorithm [Shi & Tomasi 1994]
- *Scale-Invariant Feature Transform* algorithm [Lowe 1999, Lowe 2004*a*]
- *Maximally Stable Extremal Regions* algorithm [Matas et al. 2002, Mikolajczyk et al. 2005, Nister & Stewenius 2008, Obdrzalek et al. 2009]
- *Features from Accelerated Segment Test* algorithm [Rosten & Drummond 2005]
- *Speeded-Up Robust Features* algorithm [Bay, Tuytelaars & Van Gool 2006, Bay et al. 2008, Bradski & Kaehler 2008]
- *Binary Robust Invariant Scalable Keypoints* algorithm [Leutenegger, Chli & Siegwart 2011].

The above feature detection, extraction and matching algorithms, except for the patented *Scale-Invariant Feature Transform* (SIFT) algorithm [Lowe 2004*b*], are implemented in the *Computer Vision System* and *Image Processing* toolboxes of MATLAB®.

3 Imagery datasets

Three datasets of airborne imagery were provided by DRDC for the purpose of software evaluation and benchmarking.

Figure 1: Typical image of Dataset A

The first dataset ("Dataset A") contains 302 mid-wavelength infra-red (MWIR) airborne images of a scene in the Ottawa (Canada) area. A meta-data spreadsheet with poses of cameras and a high-resolution Computer-Aided Design (CAD) model of the building obtained by terrestrial non-contact 3D imaging systems (laser scanner) were also provided by DRDC [Beraldin & Coumoyer 2014]. The images were 480-by-640 pixels in size. A typical MWIR image is shown in Figure 1.

Figure 2: Typical image of Dataset B

The second dataset ("Dataset B") contains 71 airborne images of a scene containing a church. The texture of the buildings was altered, but the rest of the scene was intact. A meta-data spreadsheet with poses of cameras was also provided by DRDC. The images were 1280-by-720 pixels in size. One of them is shown in Figure 2.

Figure 3: Typical image of Dataset C

The third dataset ("Dataset C") contains 50 airborne images of a scene containing a small farm. The images were 1000-by-1000 pixels in size. One of them is shown in Figure 3.

In addition to the real-world imagery provided by DRDC, the fourth dataset ("Dataset D")

Figure 4: Typical image of Dataset D

was evaluated, which consisted of 18 synthetic MWIR images generated by VIRSuite. This software was developed in the DST Group [Swierkowski et al. 2014]. The synthetic images were 1024-by-1024 pixels in size. One of them is shown in Figure 4.

4 Evaluation of the MATLAB toolbox

The latest releases of the *Image Processing* and *Computer Vision System* toolboxes support the above-mentioned feature detection, extraction and matching algorithms except for the SIFT algorithm. These algorithm implementations with default settings were evaluated on the four imagery datasets. All the images in each dataset were processed, and the average numbers of detected keypoints and putative matches calculated. The results are summarized in Tables 1–4, respectively.

The *Minimum Eigenvalue* algorithm detected the greatest number of features on average. However, the ratios of the average numbers of putative matches and detected keypoints were considerably smaller. On the other hand, the corresponding ratios obtained by the *Speeded-Up Robust Features* (SURF) algorithm were quite high, though the average numbers of detected keypoints were moderate. This indicates that the SURF descriptors provide better identification of the detected features.

The keypoints from first two images from each dataset, detected by the *Minimum Eigenvalue* algorithm, and their putative matches are shown in Figures A1–A4, respectively. It is

Table 1: Detected keypoints and putative matches in the images of Dataset A

Feature detection algorithm	**Detected keypoints**	**Putative matches**
Combined Corner and Edge Detector	98	63
Minimum Eigenvalue	961	556
Maximally Stable Extremal Regions	419	232
Features from Accelerated Segment Test	133	50
Speeded-Up Robust Features	477	359
Binary Robust Invariant Scalable Keypoints	141	23

Table 2: Detected keypoints and putative matches in the images of Dataset B

Feature detection algorithm	**Detected keypoints**	**Putative matches**
Combined Corner and Edge Detector	227	60
Minimum Eigenvalue	1918	296
Maximally Stable Extremal Regions	466	143
Features from Accelerated Segment Test	272	49
Speeded-Up Robust Features	713	340
Binary Robust Invariant Scalable Keypoints	256	19

Table 3: Detected keypoints and putative matches in the images of Dataset C

Feature detection algorithm	**Detected keypoints**	**Putative matches**
Combined Corner and Edge Detector	1202	191
Minimum Eigenvalue	9608	383
Maximally Stable Extremal Regions	559	156
Features from Accelerated Segment Test	984	112
Speeded-Up Robust Features	991	377
Binary Robust Invariant Scalable Keypoints	548	22

Table 4: Detected keypoints and putative matches in the images of Dataset D

Feature detection algorithm	**Detected keypoints**	**Putative matches**
Combined Corner and Edge Detector	1440	6
Minimum Eigenvalue	6776	10
Maximally Stable Extremal Regions	260	12
Features from Accelerated Segment Test	134	1
Speeded-Up Robust Features	142	8
Binary Robust Invariant Scalable Keypoints	92	0

seen that the number of matched features in the synthetic images is very small as compared with the number of detected features. The variation of default control parameters, though provided more putative matches, did not help much as the additional matched features were outliers.

5 Evaluation of VisualSFM

Several methodologies have been employed for the evaluation of 3D target reconstruction software, which are based on various types of metrics [Cavegn et al. 2014, Nex et al. 2015, Ballabeni et al. 2015, Nikolov & Madsen 2016]. In this report we did not compare reconstructed objects with 'ground-truth' data in detail, so only visual recognition of a scene from the reconstructed point cloud was applied.

VisualSFM developed by Changchang Wu [Wu 2013, Wu et al. 2011] can be downloaded from http://ccwu.me/vsfm/. This software uses the SIFT algorithm to detect feature points, and its performance depends on the Graphics Processing Unit (GPU) specification.

VisualSFM was applied to the images of Datasets A-D. The reconstructed point clouds and camera poses are shown in Figures A5–A8, respectively. Camera poses are not seen well in the figures. The graphical user interface of VisualSFM allows the user to scale, rotate and translate the reconstructed scene. In general, this provides a means to visually analyse the obtained model.

The visual examination of the obtained point clouds demonstrated that the scenes were reconstructed fairly well from the airborne images of Datasets A-C. Also, the number of the reconstructed camera poses match the number of images. However, the synthetic MWIR images did not provide a sufficient number of points. Moreover, only two cameras out of 18 were reconstructed by VisualSFM.

Though future enhancement of the VisualSFM software is expected in the view of the publication [Zheng & Wu 2015], it is very unlikely that the source code will be released. This motivated us to develop an in-house code to meet future requirements.

6 Description of the developed MATLAB code

The developed MATLAB code implements several multi-view reconstruction algorithms from [Hartley & Zisserman 2003], and currently relies on the *Image Processing* and *Computer Vision System* toolboxes of MATLAB to detect, extract and match features.

The main menu of the code is shown in Figure 5. A dataset of imagery has to be loaded first. Images can be in any standard format supported by the *Image Processing* toolbox. If camera poses are available, the user can create an XML metadata file called `metadata.xml` file whose sample fragment is shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<metadata>
   <image file="image001.jpg" x="303086" y="5.2002e+06" z="594.284"/>
   <image file="image002.jpg" x="303101" y="5.20027e+06" z="594.284"/>
   <image file="image003.jpg" x="303105" y="5.20035e+06" z="597.017"/>
   <image file="image004.jpg" x="303103" y="5.20042e+06" z="602.179"/>
   <image file="image005.jpg" x="303097" y="5.20049e+06" z="606.734"/>
   <image file="image006.jpg" x="303089" y="5.20056e+06" z="608.556"/>
   <image file="image007.jpg" x="303078" y="5.20063e+06" z="608.556"/>
```

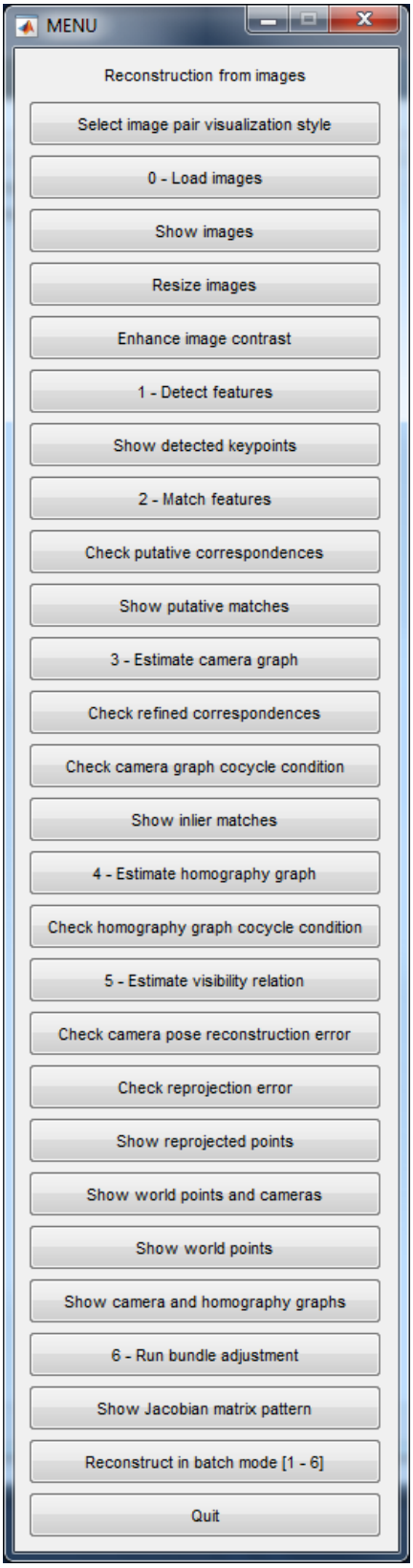


Figure 5: The main menu of the MATLAB application for target reconstruction

```
    <image file="image008.jpg" x="303064" y="5.2007e+06" z="610.074"/>
    <image file="image009.jpg" x="303044" y="5.20077e+06" z="610.378"/>
    <image file="image010.jpg" x="303018" y="5.20083e+06" z="608.556"/>
    <image file="image011.jpg" x="302989" y="5.2009e+06" z="605.216"/>
    <image file="image012.jpg" x="302955" y="5.20095e+06" z="600.964"/>
    <image file="image013.jpg" x="302917" y="5.20101e+06" z="598.839"/>
    <image file="image014.jpg" x="302874" y="5.20107e+06" z="597.624"/>
    <image file="image015.jpg" x="302829" y="5.20112e+06" z="594.891"/>
    <image file="image016.jpg" x="302779" y="5.20116e+06" z="594.284"/>
    <image file="image017.jpg" x="302725" y="5.20121e+06" z="594.588"/>
    ...
</metadata>
```

and place it in the folder where the dataset images are located. The `file` and `x`, `y`, `z` attributes of each `<image>` element must specify the image file name and the Cartesian coordinates of the camera corresponding to the taken image, respectively. This optional information is primarily used for bringing the reconstructed model to the reference frame of the cameras for better visualization, therefore camera orientation is ignored. Since this reference frame is defined by at least five cameras in general configuration, the user will be prompted to load the metadata file if five or more images are selected. Then the loaded images can be optionally resized and their contrast enhanced.

After a dataset of imagery is loaded the user can execute the following reconstruction steps:

1 Detect features

2 Match features

3 Estimate camera graph

4 Estimate homography graph

5 Estimate visibility relation

6 Run bundle adjustment.

Relevant solver control parameters can be selected at each step. Visualisation of intermediate reconstruction results is available, and each step can be repeated with adjusted solver control parameters. Alternatively, all the 6 steps can be executed in a batch mode with the default solver parameters.

In Step 1, after the feature detection algorithm is specified, all images are processed to detect and extract features. Then the detected keypoints can be viewed. The user can change the feature detection algorithm and re-run Step 1 if required.

In Step 2 the user is prompted to select the camera graph type, which specifies which image pairs to process, and then prompted to accept or change the default rejection ratio, a parameter in the match feature algorithm of the *Computer Vision System* toolbox, which is used to decide when to reject ambitious matches.

In Step 3 the user is prompted to select solver control parameters for the estimation of bifocal tensors for each edge of the camera graph. Employing the implemented RANSAC algorithm [Fischler & Bolles 1981], outliers are detected for each image pair by randomly

sampling 7-point correspondences and estimating the bifocal tensor (the fundamental matrix) using the non-linear algorithm described in [Hartley & Zisserman 2003, Page 281]. Then the bifocal tensor is re-estimated from the remaining inliers using the normalized 8-point algorithm [Hartley & Zisserman 2003, Page 282]. Finally, the *Gold Standard* method is applied to optimize the bifocal tensor by minimizing the re-projection error objective function using the implemented *Levenberg–Marquardt* solver [Levenberg 1944, Marquardt 1963]. In the end of this procedure the camera graph built on images as vertices and image pairs with sufficient number of point correspondences as edges is fully estimated. Some edges of the camera graph are deleted if an insufficient number of matched inliers are detected. After Step 3 the user can check the refined correspondences and camera graph co-cycle conditions, and view the matched inliers. The refined correspondences between matched inliers must be a bijection for each image pair, the edge of the camera graph, and the camera co-cycle condition must be satisfied for each graph cycle for consistency [Antanovskii 2017*b*]. If these conditions are not met, e.g. due to excessive noise, the user has to change the solver control parameters, such as the inlier tolerance, or choose another feature detection algorithm.

In Step 4 the user is prompted to select solver control parameters for the estimation of 3D transition homographies. The 3D homography is a projective transformation of the 3D projective space bringing locally reconstructed world points for each image pair to a global reference frame. The transition homographies are parametrized by the edges of the homography graph, which is a subgraph of the line graph [Harary 1972] of the camera graph [Antanovskii 2016*a*]. The RANSAC solver is applied to estimate the 3D homography by randomly sampling 5-point correspondences, followed by minimizing the re-projection error with respect to the corresponding local camera maps using the Levenberg–Marquardt solver. If the homography graph has cycles, the co-cycle conditions have to be satisfied [Antanovskii 2016*a*]. However, in the presence of significant noise in images, the transition homographies may not satisfy the co-cycle conditions accurately enough.

In Step 5 the user is prompted to select a uniqueness tolerance for the estimation of the visibility relation [Antanovskii 2016*b*]. First, camera maps are computed in a global coordinate system from the transition homographies and local canonical camera matrices. The locally reconstructed world points are brought to the global coordinate system at the same time. World points, reconstructed from different views, invariably contain duplicates which have to be eliminated. The uniqueness tolerance is the threshold for merging potentially duplicate points. A metric between two 3D points is defined as the maximum of the distances of projected image points (e.g. in pixels) averaged over a set of cameras. This metric may not be a proper distance, because it can vanish for different points when the set of cameras is degenerate. For example, this situation always occurs for a single camera. However, in the general configuration of at least two cameras, this metric has all the properties of a distance. This approach is promising as it easily copes with the common situation when a 3D point disappears from a view and then re-appears again. After Step 5 the user can check the camera pose reconstruction error, if the matadata file was loaded, check the re-projection error, view the re-projected keypoints in images, view the world points with or without camera positions, and view the camera and homography graphs. A random 3D homography can be applied from the user interface to change the projective view. There are 15 degrees of freedom for the selection of an appropriate homography, which are not intuitive as opposed to object rotation. So, a random homography is easier to apply for the visualization purpose, which may eventually make the reconstructed scene more recognizable after a few attempts.

In Step 6 the user is prompted to select solver control parameters for *Bundle Adjustment*. The whole set of world points without duplicates and camera maps is optimized by minimizing the global re-projection error using the Levenberg–Marquardt solver. A sparse storage for the associated Jacobian matrices is used in the implementation of the bundle adjustment. After Step 6 the pattern of the sparse Jacobian matrix can be visualized.

7 Discussion

Four imagery datasets were evaluated with the *Computer Vision System* and *Image Processing* toolboxes of MATLAB, and with VisualSFM. It was revealed that the real-world imagery provided reasonably good reconstruction of the structures and camera poses, whereas the synthetic images generated by VIRSuite performed poorly. The latter issue will be addressed in the future.

The user interface of developed MATLAB code has been described, which is currently based on the toolboxes of MATLAB [Antanovskii 2016*a*, Antanovskii 2017*b*]. The code is thoroughly tested in unit tests and partially validated against the imagery datasets. The integration of the implemented SIFT algorithm [Antanovskii 2017*a*] with the MATLAB code will be the subject of a separate publication. Another important extension of the code to be addressed in the future is its integration with LADAR sensing data.

Acknowledgements

The author is grateful to Defence Research and Development Canada for providing the airborne imagery for code validation and benchmarking. Valuable discussion with Dr Leszek Swierkowski from the Defence Science and Technology Group is much appreciated.

8 References

Antanovskii, L. K. (2014) *Implementation of geometric algebra in MATLAB® with applications*, Technical Report DSTO–TR–3021, DSTO, Edinburgh, Australia.

Antanovskii, L. K. (2016*a*) *Mathematical aspects of computer vision*, Technical Report DST-Group-TR-3214, DST Group, Edinburgh, Australia.

Antanovskii, L. K. (2016*b*) *Projective reconstruction of world points and camera matrices from a sequence of images with MATLAB®*, Technical Report DST-Group-TR-3213, DST Group, Edinburgh, Australia.

Antanovskii, L. K. (2017*a*) *Implementation of the Scale Invariant Feature Transform algorithm in MATLAB*, Technical Report DST-Group–TR–3347, DST Group, Edinburgh, Australia.

Antanovskii, L. K. (2017*b*) *Sparse reconstruction of a scene and camera poses from the scene images with MATLAB*, Technical Report DST-Group–TR–3346, DST Group, Edinburgh, Australia.

Ballabeni, A., Apollonio, F. I., Gaiani, M. & Remondino, F. (2015) Advances in image pre-processing to improve automated 3D reconstruction, *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **XL-5/W4**, 315–323.

Bay, H., Ess, A., Tuytelaars, T. & Van Gool, L. (2008) Speeded-up robust features (SURF), *Computer Vision and Image Understanding* **110**(3), 346–359.

Bay, H., Tuytelaars, T. & Van Gool, L. (2006) SURF: Speeded up robust features, *in Proc. 9th European Conf. Computer Vision*.

Beraldin, J.-A. & Coumoyer, L. (2014) *Acquisition of a dense 3D point cloud of the exterior walls and the court yards of the NRC Sussex Building in Ottawa*, Technical Report NRC-MSS-MM, National Research Council Canada.

Bradski, G. & Kaehler, A. (2008) *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly, Sebastopol, CA.

Cavegn, S., Haala, N., Nebiker, S., Rothermel, M. & Tutzauer, P. (2014) Benchmarking high density image matching for oblique airborne imagery, *in Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, Vol. XL-3, Zürich, Switzerland, pp. 45–52.

Fischler, M. A. & Bolles, R. C. (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. Assoc. Comp. Mach.* **24**(6), 381–395.

Harary, F. (1972) *Graph Theory*, Addison-Wesley, Reading, MA.

Harris, C. & Stephens, M. (1988) A combined corner and edge detector, *in Proc. 4th Alvey Vision Conf.*, pp. 147–151.

Hartley, R. & Zisserman, A. (2003) *Multiple View Geometry in Computer Vision*, 2nd edn, Cambridge University Press, Cambridge.

Leutenegger, S., Chli, M. & Siegwart, R. (2011) BRISK: Binary robust invariant scalable keypoints, *in Proc. IEEE Int. Conf. Computer Vision*.

Levenberg, K. (1944) A method for the solution of certain non-linear problems in least squares, *Quart. Appl. Math.* **2**, 164–168.

Lowe, D. G. (1999) Object recognition from local scale-invariant features, *in Proc. Int. Conf. Computer Vision*, Vol. 2, pp. 1150–1157.

Lowe, D. G. (2004*a*) Distinctive image features from scale-invariant keypoints, *in Int. J. Computer Vision*, Vol. 60, pp. 91–110.

Lowe, D. G. (2004*b*) Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image. US Patent 6,711,293.

Marquardt, D. (1963) An algorithm for least-squares estimation of nonlinear parameters, *SIAM J. Appl. Math.* **11**(2), 431–441.

Matas, J., Chum, O., Urba, M. & Pajdla, T. (2002) Robust wide baseline stereo from maximally stable extremal regions, *in Proc. British Machine Vision Conf.*, pp. 384–396.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Kadir, T. & Van Gool, L. (2005) A comparison of affine region detectors, *Int. J. Computer Vision* **65**(1–2), 43–72.

Nex, F., Gerke, M., Remondino, F., Przybilla, H.-J., Bäumker, M. & Zurhorst, A. (2015) ISPRS benchmark for multi-platform photogrammetry, *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* **II-3/W4**, 135–142.

Nikolov, I. & Madsen, C. (2016) Benchmarking close-range structure from motion 3D reconstruction software under varying capturing conditions, *in Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection: 6th International Conference, EuroMed 2016, Nicosia, Cyprus, October 31 – November 5, 2016*, Vol. XXVIII, Springer International Publishing, Cham, pp. 15–26.

Nister, D. & Stewenius, H. (2008) Linear time maximally stable extremal regions, *in Proc. 10th European Conf. Computer Vision*, Vol. 5303 of *Lecture Notes in Computer Science*, Marseille, France, pp. 183–196.

Obdrzalek, D., Basovnik, S., Mach, L. & Mikulik, A. (2009) Detecting scene elements using maximally stable colour regions, *in Communications in Computer and Information Science*, Vol. 82, La Ferte-Bernard, France, pp. 107–115.

Rosten, E. & Drummond, T. (2005) Fusing points and lines for high performance tracking, *in Proc. IEEE Int. Conf. Computer Vision*, Vol. 2, pp. 1508–1511.

Shi, J. & Tomasi, C. (1994) Good features to track, *in Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 593–600.

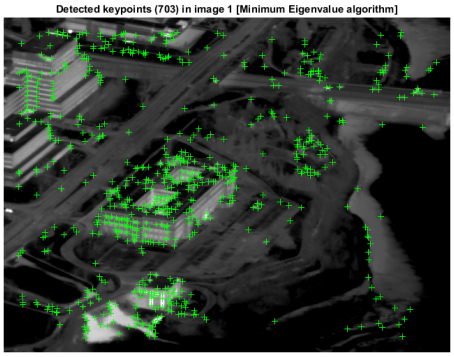
Swierkowski, L., Christie, C. L., Antanovskii, L. K. & Gouthas, E. (2014) Real-time scene and signature generation for ladar and imaging sensors, *in Proc. SPIE 9071 Infrared Imaging System: Design, Analysis, Modeling, and Testing XXV*, Vol. 90711E, Baltimore, USA.

Wu, C. (2013) Towards linear-time incremental structure from motion, *in IEEE Int. Conf. 3D Vision*, pp. 127–134.

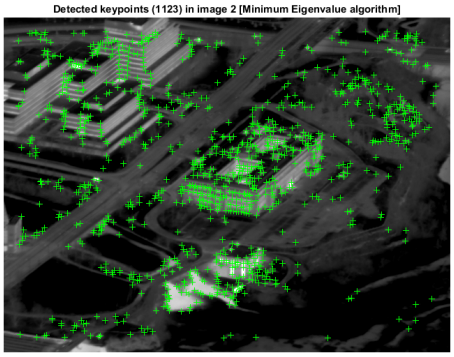
Wu, C., Agarwal, S., Curless, B. & Seitz, S. M. (2011) Multicore bundle adjustment, *in IEEE Conf. Computer Vision and Pattern Recognition*, pp. 3057–3064.

Zheng, E. & Wu, C. (2015) Structure from motion using structure-less resection, *in IEEE Int. Conf. Computer Vision (ICCV)*, pp. 2075–2083.

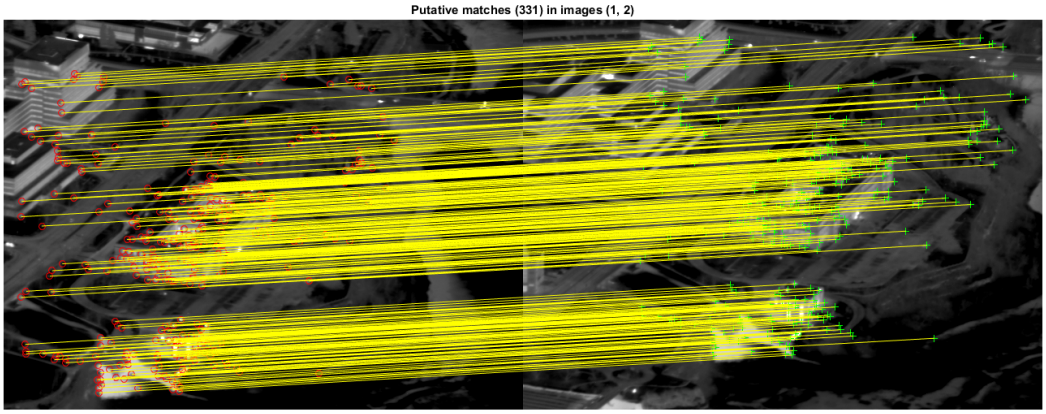
Appendix A: Simulation results



(i) Detected features in Image 1



(ii) Detected features in Image 2

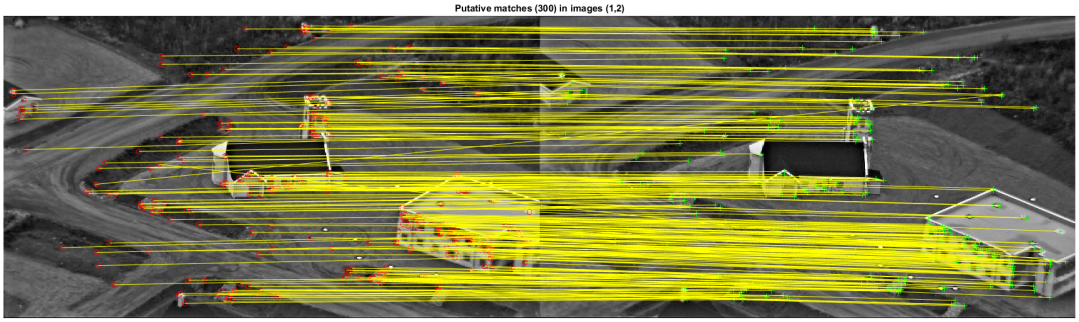


(iii) Putative matches

Figure A1: Detected and matched features in two images of Dataset A

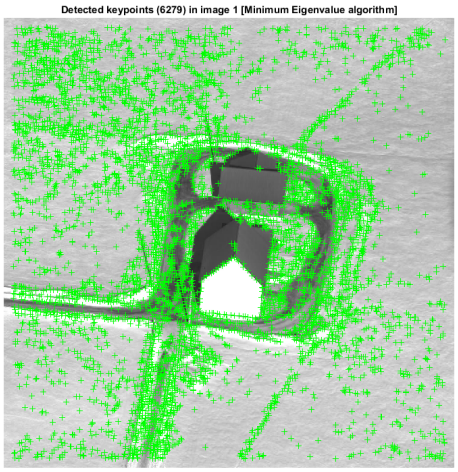
(i) Detected features in Image 1

(ii) Detected features in Image 2

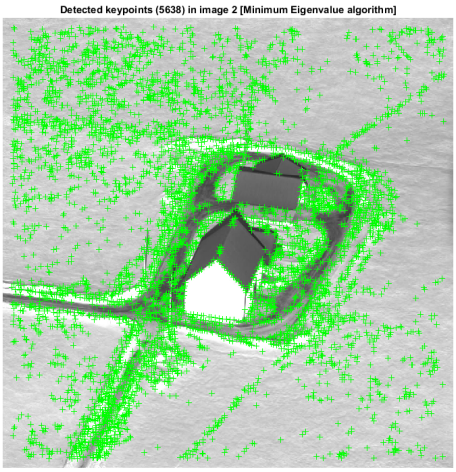


(iii) Putative matches

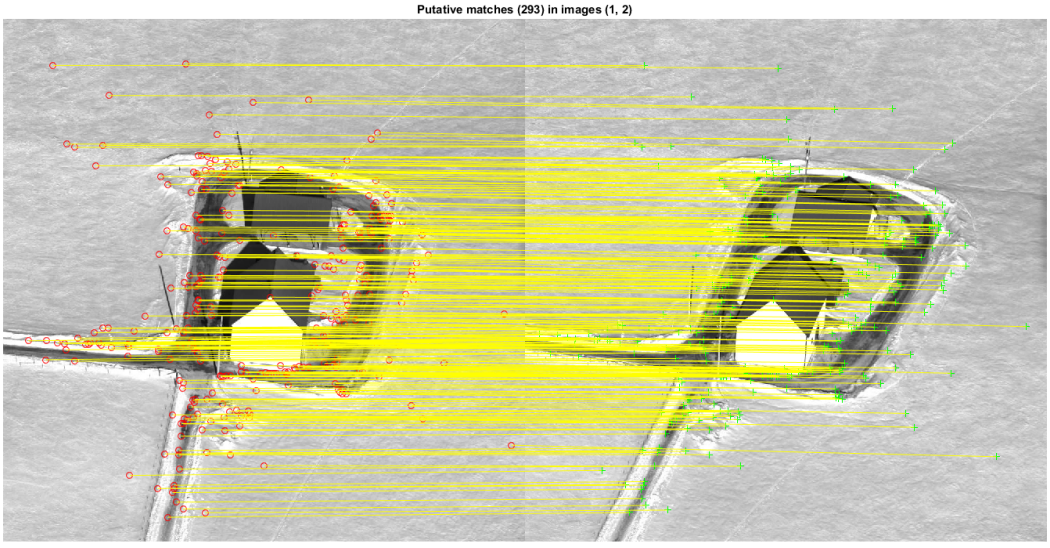
Figure A2: Detected and matched features in two images of Dataset B



(i) Detected features in Image 1

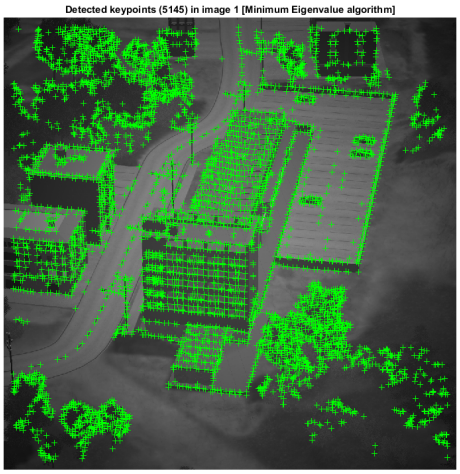


(ii) Detected features in Image 2

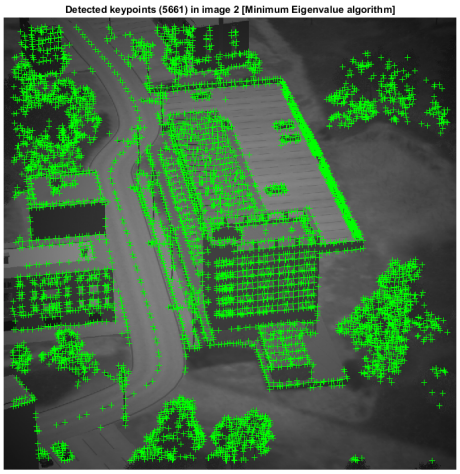


(iii) Putative matches

Figure A3: Detected and matched features in two images of Dataset C



(i) Detected features in Image 1



(ii) Detected features in Image 2

(iii) Putative matches

Figure A4: Detected and matched features in two images of Dataset D

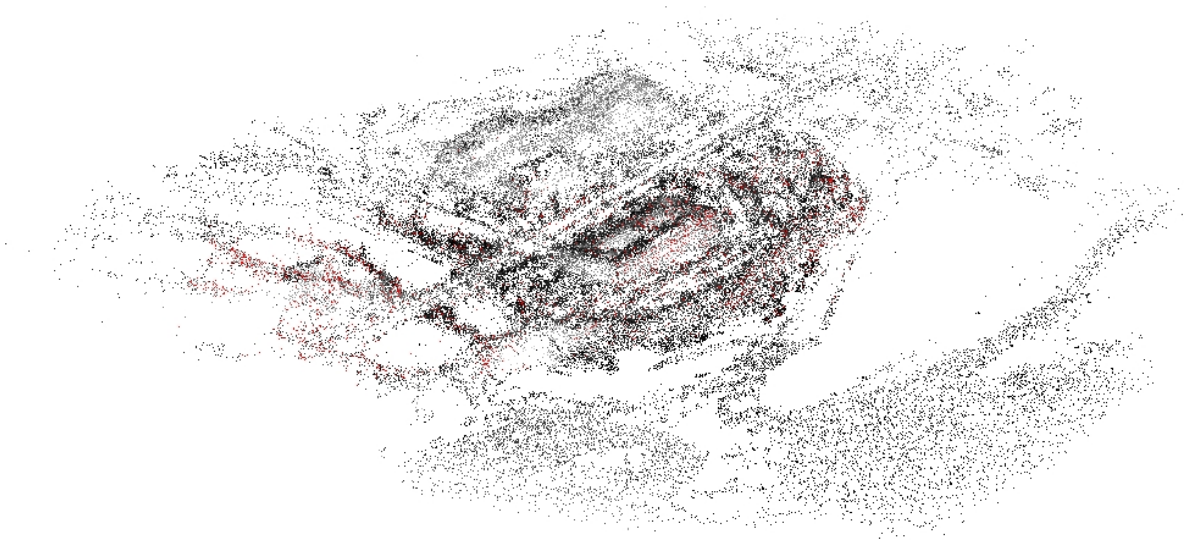


Figure A5: Point cloud and camera poses reconstructed from 302 images of Dataset A

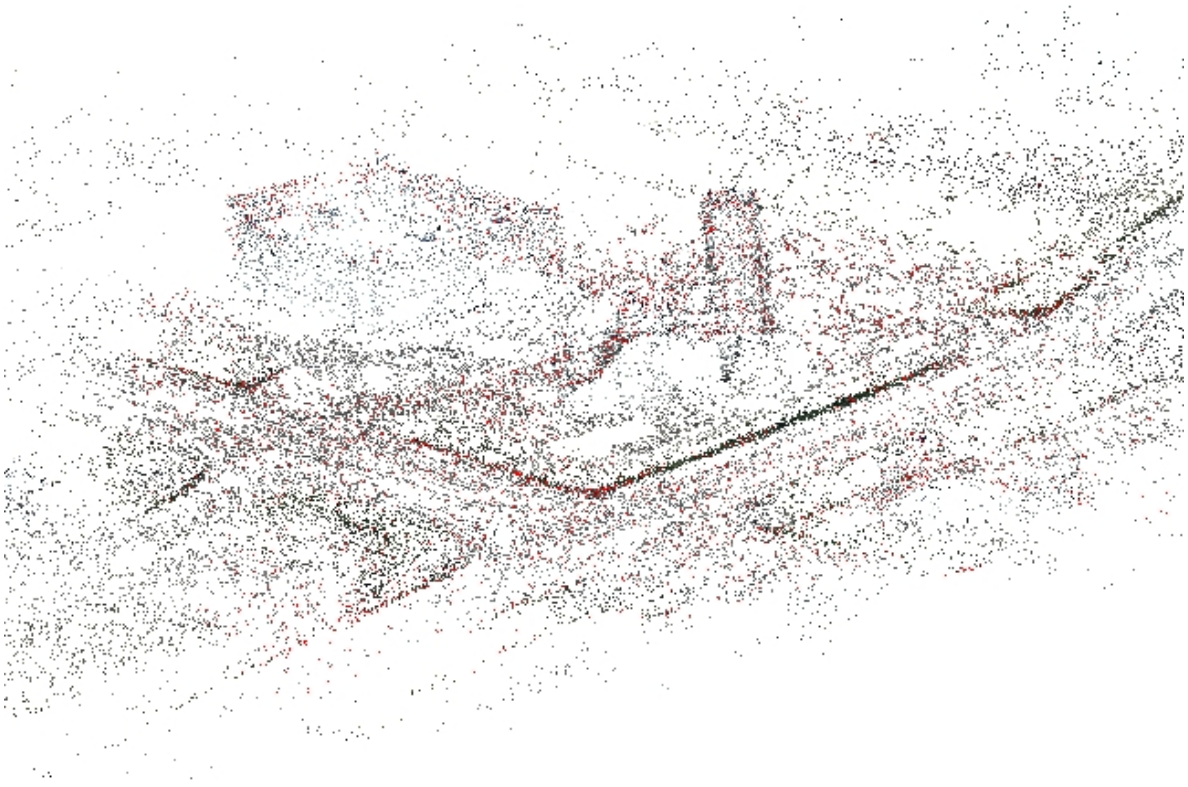


Figure A6: Point cloud and camera poses reconstructed from 71 images of Dataset B

Figure A7: Point cloud and camera poses reconstructed from 50 images of Dataset C

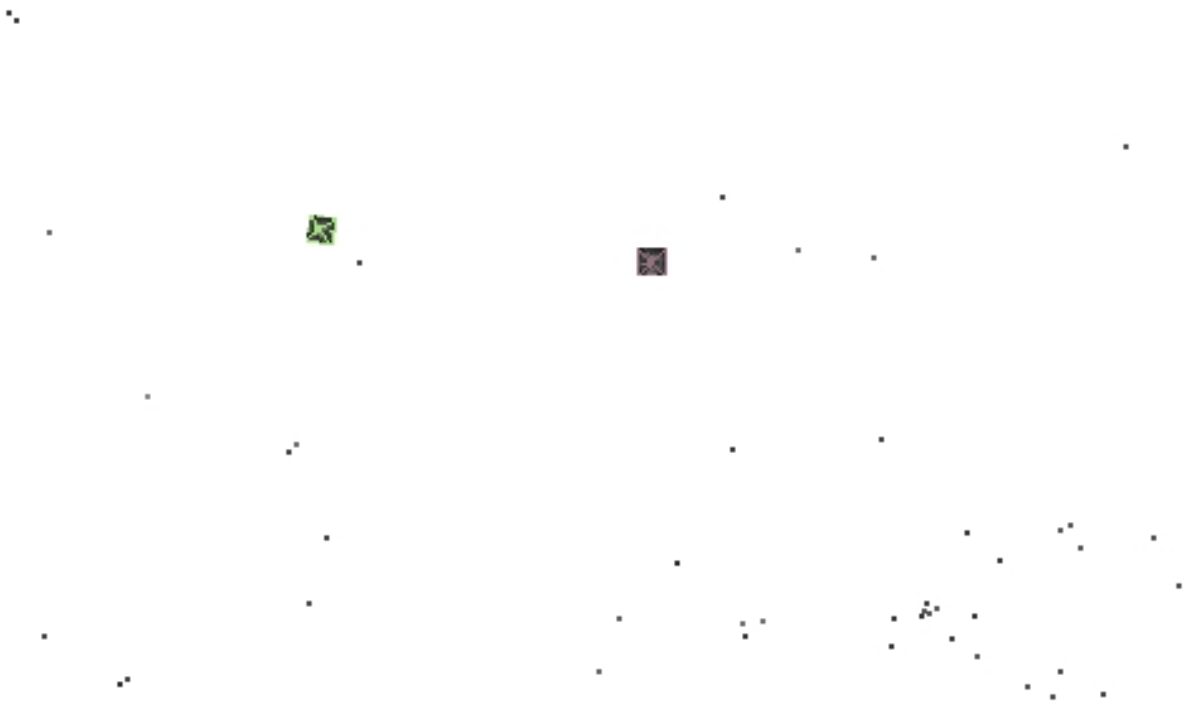


Figure A8: Point cloud and camera poses reconstructed from 18 images of Dataset D

UNCLASSIFIED

DEFENCE SCIENCE AND TECHNOLOGY GROUP DOCUMENT CONTROL DATA			1. DLM/CAVEAT (OF DOCUMENT)
2. TITLE Evaluation of Available Software for Reconstruction of a Structure from its Imagery		3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)	
4. AUTHOR Leonid K Antanovskii		5. CORPORATE AUTHOR Defence Science and Technology Group PO Box 1500 Edinburgh, South Australia 5111, Australia	
6a. DST Group NUMBER DST-Group–TR–3356	6b. AR NUMBER 016-831	6c. TYPE OF REPORT Technical Report	7. DOCUMENT DATE April, 2017
8. Objective ID AV12700547	9. TASK NUMBER AIR07/213	10. TASK SPONSOR RAAF Air Combat Group	
13. DST Group Publications Repository http://dsto.defence.gov.au/		14. RELEASE AUTHORITY Chief, Weapons and Combat Systems Division	

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SOUTH AUSTRALIA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS

No Limitations

18. RESEARCH LIBRARY THESAURUS

Science, Mathematics, Algorithms, Computer Vision, Structure Reconstruction

19. ABSTRACT

In this report the *Computer Vision System* toolbox of MATLAB® and the *Visual Structure from Motion* software are evaluated on three datasets of airborne imagery, provided by Defence Research and Development Canada, and on a dataset of synthetic imagery generated by the VIRSuite software developed in the Defence Science and Technology Group. The user interface for a developed code for structure reconstruction, based on the feature detection algorithms of the MATLAB toolbox, is described.

UNCLASSIFIED